

# The Complete SC-Invariant Affine Automorphisms of Polar Codes

Zicheng Ye<sup>†‡</sup>, Yuan Li<sup>†‡\*</sup>, Huazi Zhang<sup>\*</sup>, Rong Li<sup>\*</sup>, Jun Wang<sup>\*</sup>, Guiying Yan<sup>†‡</sup>, and Zhiming Ma<sup>†‡</sup>

<sup>†</sup> University of Chinese Academy of Sciences

<sup>‡</sup> Academy of Mathematics and Systems Science, CAS

<sup>\*</sup> Huawei Technologies Co. Ltd.

Email: {yezicheng, liyuan2018}@amss.ac.cn, {zhanghuazi, lirongone.li, justin.wangjun}@huawei.com, yangy@amss.ac.cn, mazm@amt.ac.cn

**Abstract**—Automorphism ensemble (AE) decoding for polar codes was proposed by decoding permuted codewords with successive cancellation (SC) decoders in parallel and hence has lower latency compared to that of successive cancellation list (SCL) decoding. However, some automorphisms are SC-invariant, thus are redundant in AE decoding. In this paper, we find a necessary and sufficient condition related to the block lower-triangular structure of transformation matrices to identify SC-invariant automorphisms. Furthermore, we provide an algorithm to determine the complete SC-invariant affine automorphisms under a specific polar code construction.

## I. INTRODUCTION

Polar codes [1] are proved to asymptotically achieve capacity on discrete binary memoryless symmetric (BMS) channels under SC decoding. To enhance the finite-length performance, SCL decoding was proposed in [2]. Moreover, cyclic redundancy check (CRC)-aided polar codes [3] achieve outstanding performance at short to moderate block lengths.

A substantial part of SCL decoding complexity and latency is related to path management, i.e., sorting and pruning paths according to path metric (PM). In order to reduce the latency, decoding under stage permutations on the factor graph was proposed in [4]. In [5], AE decoding utilized more SC-variant automorphisms instead of only stage permutations to enhance error correcting performance. A key step in AE decoding is the identification and avoidance of SC-invariant automorphisms, which produce duplicate decoding results. The automorphisms formed by lower-triangular affine (LTA) transformations [6] were proved to be SC-invariant [5]. In [7] and [8], the block lower-triangular affine (BLTA) group was proved to be the complete affine automorphism group of polar codes. BLTA transformations showed better performance under AE decoding [7] [9] [10]. In [10], affine automorphism group was classified into equivalent classes, where each class will yield the same SC decoding result. Therefore selecting at most one automorphism from each equivalent class guarantees SC-variance. In contrast of AE decoding, some other applications require SC-invariant automorphisms. In [11],  $\frac{n}{4}$ -cyclic shift permutations, which are SC-invariant, were proposed for implicit timing indication in Physical Broadcasting Channel (PBCH). In both applications, identifying SC-invariant automorphisms is a key step.

Some previous works attempt to identify SC-invariant affine automorphisms for general polar codes [5] [10]. However, given a specific code construction, SC-invariant automorphisms can not be completely found in [5], [10]. In this paper, we identify and prove the complete SC-invariant affine automorphisms. For example, as shown in Table 1 of section IV, the number of the complete SC-invariant affine automorphisms for (256,128) polar code is  $21 \times 2^{28}$  but only  $3 \times 2^{28}$  of them are founded in [10].

The rest of this paper is organized as follows. In section II, we review polar codes and automorphism group. In section III, we provide a low complexity algorithm to distinguish SC-invariant affine automorphisms. We further prove SC-invariant affine automorphism group is also of the form BLTA and provide an algorithm to determine it. In section IV, simulation results show distinguishing SC-invariant automorphisms can reduce redundancy in AE decoding. Finally, we draw some conclusions in section V.

## II. PRELIMINARIES

### A. Polar codes as monomial codes

Let  $F = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$  and  $G_m = F^{\otimes m}$ , where  $m$  is the code dimension. A polar code  $(n = 2^m, K)$  is generated by selecting  $K$  rows of  $G_m$ . The set  $\mathcal{I} \subseteq \{0, 1, \dots, n-1\}$  of indices of selected rows is the information set, and  $\mathcal{F} = \mathcal{I}^c$  is the frozen set. Denote the polar code with information set  $\mathcal{I}$  by  $C(\mathcal{I})$ .

Polar codes can be described as monomial codes [6]. The monomial set is

$$\mathcal{M} = \{x_1^{g_1} \dots x_m^{g_m} | (g_1, \dots, g_m)^T \in \mathbb{F}_2^m\},$$

and the evaluation of  $g \in \mathcal{M}$  is

$$\text{eval}(g) = (g(u))_{u \in \mathbb{F}_2^m}.$$

Then each row of  $G_m$  can be represented by  $\text{eval}(g)$  for some  $g \in \mathcal{M}$ . For example, assume  $z \in \{0, 1, \dots, 2^m - 1\}$ , there is a unique binary representation  $a = (a_1, \dots, a_m)^T$  of  $2^m - z - 1$ , where  $a_1$  is the least significant bit, such that

$$\sum_{i=1}^m 2^{i-1} (1 - a_i) = z.$$

Then the evaluation of monomial  $\text{eval}(x_1^{a_1} \dots x_m^{a_m})$  is exactly the  $(2^m - z - 1)$ -th row of  $G_m$ . Therefore, the information set  $\mathcal{I}$  can be regarded as a subset of  $\{0, \dots, n-1\}$  or a subset of  $\mathcal{M}$ . As seen, the three representations, i.e., the number  $z$ , the binary representation of  $2^m - 1 - z = (a_1, \dots, a_m)^T$  and the corresponding monomial  $x_1^{a_1} \dots x_m^{a_m}$  all refer to the same thing.

Two monomials of the same degree are ordered as  $x_{i_1} \dots x_{i_t} \preceq x_{j_1} \dots x_{j_t}$  if and only if  $i_l \leq j_l$  for all  $l \in \{1, \dots, t\}$ , where we assume  $i_1 < \dots < i_t$  and  $j_1 < \dots < j_t$ . This partial order is extended to monomials with different degrees through divisibility, namely  $f \preceq g$  if and only if there is a divisor  $g'$  of  $g$  such that  $f \preceq g'$ .

An information set  $\mathcal{I} \subseteq \mathcal{M}$  is decreasing if  $\forall g \preceq f$  and  $f \in \mathcal{I}$  we have  $g \in \mathcal{I}$ . A decreasing monomial code  $C(\mathcal{I})$  is a monomial code with a decreasing information set  $\mathcal{I}$ . If the information set is selected according to the Bhattacharyya parameter, polar codes will be decreasing monomial codes [6], [12]. In this way, polar codes can be generated by  $\mathcal{I}_{\min}$ , where the information set is the smallest decreasing set containing  $\mathcal{I}_{\min}$ . From now on, we always suppose  $\mathcal{I}$  is decreasing.

### B. Affine automorphism group

Let  $C$  be a decreasing monomial code with length  $n$ . A permutation  $\pi$  in the symmetric group  $\text{Sym}(n)$  is an automorphism of  $C$  if for any codeword  $c = (c_0, \dots, c_{n-1}) \in C$ ,  $\pi(c) = (c_{\pi(0)}, \dots, c_{\pi(n-1)}) \in C$ . The automorphism group  $\text{Aut}(C)$  is the subgroup of  $\text{Sym}(n)$  containing all automorphisms of  $C$ .

Let  $M$  be an  $m \times m$  binary invertible matrix and  $b$  be a length- $m$  binary column vector. The affine transformation  $(M, b)$  permutes  $a \in \mathbb{F}_2^m$  to  $Ma + b$ .

A matrix  $M$  is lower-triangular if  $M(i, i) = 1$  and  $M(i, j) = 0$  for all  $j > i$ . The LTA group is the group of all affine transformations  $(M, b)$  where  $M$  is lower-triangular. Similarly, a matrix  $M$  is upper-triangular if  $M(i, i) = 1$  and  $M(i, j) = 0$  for all  $j < i$ .

BLTA( $[s_1, \dots, s_l]$ ) is a BLTA group of all affine transformations  $(M, b)$  where  $M$  can be written as a block matrix of the following form

$$\begin{bmatrix} B_{1,1} & 0 & \cdots & 0 \\ B_{2,1} & B_{2,2} & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ B_{l,1} & B_{l,2} & \cdots & B_{l,l} \end{bmatrix}, \quad (1)$$

where  $B_{i,i}$  are full-rank  $s_i \times s_i$  matrices. BLTA equals the complete automorphisms of decreasing polar codes that can be formulated as affine transformations [8].

### C. Successive cancellation decoding

Let  $L_{i,t}$  be the log likelihood ratio (LLR) of the  $i$ -th node at stage  $t$  and  $L_{i,m}$  be the received LLRs from channels.  $L_{i,t}$  are propagated from stage  $t+1$  according to

$$L_{i,t} = f(L_{i,t+1}, L_{i+2^t, t+1}) = \log \left( \frac{e^{L_{i,t+1} + L_{i+2^t, t+1}} + 1}{e^{L_{i,t+1}} + e^{L_{i+2^t, t+1}}} \right);$$

$$\begin{aligned} L_{i+2^t, t} &= g(u_{i,t}, L_{i,t+1}, L_{i+2^t, t+1}) \\ &= (-1)^{u_{i,t}} L_{i,t+1} + L_{i+2^t, t+1}. \end{aligned}$$

At stage 0, we have

$$u_{i,0} = \begin{cases} 0, & \text{if } i \in \mathcal{F}; \\ 0, & \text{if } i \in \mathcal{I}, L_{i,0} \geq 0; \\ 1, & \text{if } i \in \mathcal{I}, L_{i,0} < 0. \end{cases}$$

Then hard decisions  $u_{i,t}$  are propagated from stage  $t-1$  according to

$$u_{i,t} = u_{i,t-1} \oplus u_{i+2^t, t-1};$$

$$u_{i+2^t, t} = u_{i+2^t, t-1}.$$

where  $\oplus$  means the addition modulo 2.

Let  $\text{SC}_{\mathcal{I}} : \mathbb{R}^n \rightarrow \mathbb{F}_2^n$  map the received LLR vector  $y = (L_{i,m})_{i \in \{0, \dots, n-1\}} \in \mathbb{R}^n$  to the SC decoding result  $(u_{i,m})_{i \in \{0, \dots, n-1\}} = \text{SC}_{\mathcal{I}}(y) \in C(\mathcal{I})$ .

### D. Automorphism ensemble decoding

Let  $\pi_1, \dots, \pi_t$  be  $t$  different automorphisms of the code  $C$  and  $y \in \mathbb{R}^n$  be the received LLR vector. A list of decoders can independently decode each permuted LLR  $\pi_j(y)$ . The decoded candidate codeword of  $y$  using  $\pi_j$  is

$$\hat{x}_j = \pi_j^{-1}(\text{SC}_{\mathcal{I}}(\pi_j(y))).$$

A final decoding result is selected according to the minimum Euclidean distance rule:

$$x = \arg \min_{\hat{x}_j, j=1, \dots, t} \|\hat{x}_j - y\|.$$

For an automorphism  $\pi$  of  $C(\mathcal{I})$ , we say  $\pi$  commutes with  $\text{SC}_{\mathcal{I}}$  if for all  $y \in \mathbb{R}^n$ ,  $\text{SC}_{\mathcal{I}}(\pi(y)) = \pi(\text{SC}_{\mathcal{I}}(y))$ . If  $\pi$  commutes with  $\text{SC}_{\mathcal{I}}$ , the corresponding permuted SC decoder always outputs the same decoding result as the non-permuted SC decoder.

The automorphism  $\pi$  in LTA group is SC-invariant for  $C(\mathcal{I})$ , which means it commutes with  $\text{SC}_{\mathcal{I}}$  [5]. Moreover, in [10], two affine automorphisms  $\pi, \pi'$  are called equivalent for  $C(\mathcal{I})$ , denoted by  $\pi \sim_{\mathcal{I}} \pi'$ , if for all  $y \in \mathbb{R}^n$

$$\pi^{-1}(\text{SC}_{\mathcal{I}}(\pi(y))) = \pi'^{-1}(\text{SC}_{\mathcal{I}}(\pi'(y))).$$

The equivalence classes are defined as

$$[\pi]_{\mathcal{I}} = \{\pi' : \pi' \sim_{\mathcal{I}} \pi\}.$$

Let  $\mathbb{1}$  be identity permutation, the equivalence class  $[\mathbb{1}]_{\mathcal{I}}$  consists of the complete affine automorphisms commuting with  $\text{SC}_{\mathcal{I}}$ , which is an automorphism subgroup. The authors of [10] proved that  $\text{BLTA}([2, 1, \dots, 1]) \subseteq [\mathbb{1}]_{\mathcal{I}}$ , that is, automorphisms in  $\text{BLTA}([2, 1, \dots, 1])$  commute with  $\text{SC}_{\mathcal{I}}$  of any decreasing monomial code  $C(\mathcal{I})$  whose automorphism group includes  $\text{BLTA}([2, 1, \dots, 1])$ . A natural question arises: are these the complete SC-invariant affine automorphisms?

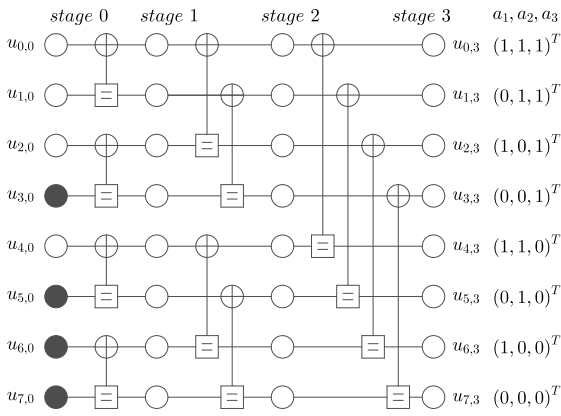


Fig. 1: The factor graph of an (8,4) polar code

### III. ANALYSIS ON SC-INVARIANT AUTOMORPHISMS

In this section, we give a necessary and sufficient condition to identify SC-invariant affine automorphisms for any specific code  $C(\mathcal{I})$ . The key technique in proofs is that the block lower-triangular structure of transformation matrix can be used to decompose the corresponding automorphism into shorter ones (detailed description is in Definition 1). Therefore,  $C(\mathcal{I})$  can be decomposed to shorter subcodes, and we can identify SC-invariant automorphisms inductively.

#### A. Notations and definitions

Let  $M$  be a full-rank matrix. We say  $M$  has the block lower-triangular structure  $s(M) = \langle s_1, \dots, s_l \rangle$  if  $M$  can be written as (1) and none of  $B_{i,i}$  can be written as a block lower-triangular matrix with more than one block. Define  $S_t = \sum_{i=1}^{t-1} s_i$  for  $2 \leq t \leq l+1$  and  $S_1 = 0$ .

Define  $[a, b]$  to be the integer set  $\{i \in \mathbb{N} | a \leq i \leq b\}$  for  $a, b \in \mathbb{N}$ , and  $M([a, b], [c, d])$  to be the corresponding submatrix of  $M$ .

The affine transformation  $(M, 0) \subseteq [\mathbf{1}]_{\mathcal{I}}$  if and only if  $(M, b) \subseteq [\mathbf{1}]_{\mathcal{I}}$  for any  $b \in \mathbb{F}_2^m$ . For convenience, define  $\varphi(M)$  to be the permutation  $(M, 0)$  which permutes  $a \in \mathbb{F}_2^m$  to  $Ma$ .

Define  $\text{Ind}_m(a_{i_1} = c_{i_1}, \dots, a_{i_t} = c_{i_t}) = \{(a_1, \dots, a_m)^T \in \mathbb{F}_2^m | a_{i_j} = c_{i_j}, \forall j = 1, \dots, t\}$  to be a set of indices whose  $i_1, \dots, i_t$ -th bits are fixed to be  $a_{i_1}, \dots, a_{i_t}$ . Define

$$\begin{aligned} & \mathcal{I}(\text{Ind}_m(a_{i_1} = c_{i_1}, \dots, a_{i_t} = c_{i_t})) \\ &= \{(a_1, \dots, a_{i_1-1}, a_{i_1+1}, \dots, a_{i_t-1}, a_{i_t+1}, \dots, a_m)^T \in \mathbb{F}_2^{m-t} | \\ & \quad (a_1, \dots, a_m)^T \in \mathcal{I}, a_{i_j} = c_{i_j}, \forall j = 1, \dots, t\} \end{aligned}$$

to be the information set of length- $2^{m-t}$  subcode consisting of indices in  $\text{Ind}_m(a_{i_1} = c_{i_1}, \dots, a_{i_t} = c_{i_t})$ .  $\mathcal{F}(\text{Ind}_m(a_{i_1} = c_{i_1}, \dots, a_{i_t} = c_{i_t}))$  is defined in the same way.

For example, the factor graph of (8,4) polar code is shown in Fig. 1, where  $\mathcal{I} = \{3, 5, 6, 7\}$ . In Fig. 1,  $\mathcal{I}(\text{Ind}_3(a_1 = 1)) = \{3\}$  (resp.  $\mathcal{I}(\text{Ind}_3(a_1 = 0)) = \{1, 2, 3\}$ ) is the information set of length-4 subcode consisting of indices that the least significant bit  $a_1$  is equal to 1 (resp. 0), i.e. the even (resp. odd) bits. Similarly,  $\mathcal{I}(\text{Ind}_3(a_3 = 1)) = \{3\}$  (resp.  $\mathcal{I}(\text{Ind}_3(a_3 =$

$0)) = \{1, 2, 3\}$ ) is the information set of length-4 subcode consisting of indices that the most significant bit  $a_3$  is equal to 1 (resp. 0), i.e. the first (resp. last) four bits. This definition simplifies the description of decomposed shorter subcodes in our proofs.

#### B. Transformations of matrices

In this subsection, we provide two lemmas on matrix transformations.

**Lemma 1 (lower-triangular transformation).** *Let  $M$  be a full-rank matrix and  $M_1, M_2$  be two lower-triangular matrices.  $\pi = \varphi(M)$  and  $\pi' = \varphi(M_1 M M_2)$  are two automorphisms of  $C(\mathcal{I})$ . Then  $\pi'$  commutes with  $SC_{\mathcal{I}}$  if and only if  $\pi$  commutes with  $SC_{\mathcal{I}}$ . Moreover,  $s(M) = s(M_1 M M_2)$ .*

*Proof.* It is from the fact that  $\varphi(M_1 M M_2) = \varphi(M_1) \varphi(M) \varphi(M_2)$  and  $\varphi(M_1), \varphi(M_2)$  and  $\varphi(M)$  all commute with  $SC_{\mathcal{I}}$ .

Let  $s(M) = \langle s_1, \dots, s_l \rangle$  and  $s(M_1 M M_2) = \langle s'_1, \dots, s'_k \rangle$ . Notice that  $L_1 M$  means adding upper row of  $M$  to lower, while  $M L_2$  means adding right column of  $M$  to left. Therefore,  $M([1, s_1], [s_1 + 1, m]) = 0$  implies  $M_1 M M_2([1, s_1], [s_1 + 1, m]) = 0$ , so  $s'_1 \leq s_1$ .

Since  $M = M_1^{-1} (M_1 M M_2) M_2^{-1}$ , similarly, we have  $s_1 \leq s'_1$ . Therefore,  $s_1 = s'_1$ . And so on,  $s(M) = s(M_1 M M_2)$ .  $\square$

**Remark 1.** Thanks to Lemma 1, we only need to investigate upper-triangular matrices since every matrix  $M$  can be transformed to an upper-triangular matrix by lower-triangular transformation while maintaining the block lower-triangular structure.

Next, we show how to decompose upper-triangular transformation by exploiting its block lower-triangular structure.

**Definition 1.** Let  $M$  be an upper-triangular matrix with  $s(M) = \langle s_1, \dots, s_l \rangle$  and  $\pi = \varphi(M)$ . We have  $M = M_1 M_2$  where

$$M_1(i, j) = \begin{cases} M(i, j), & \text{if } 1 \leq i, j \leq S_l; \\ 1, & \text{if } S_l + 1 \leq i = j \leq m; \\ 0, & \text{otherwise.} \end{cases}$$

And

$$M_2(i, j) = \begin{cases} M(i, j), & \text{if } S_l + 1 \leq i, j \leq m; \\ 1, & \text{if } 1 \leq i = j \leq S_l; \\ 0, & \text{otherwise.} \end{cases}$$

We define four permutations related to  $\pi$ :  $\pi_1 = \varphi(M_1)$ ,  $\pi_2 = \varphi(M_2)$ ,  $\tilde{\pi}_1 = \varphi(M([1, S_l], [1, S_l]))$ ,  $\tilde{\pi}_2 = \varphi(M([S_l + 1, m], [S_l + 1, m]))$ .

**Lemma 2 (Permutation Decomposition).** *Let  $M$  be an upper-triangular matrix with  $s(M) = \langle s_1, \dots, s_l \rangle$  and  $\pi = \varphi(M)$ . Let  $\pi_1, \pi_2, \tilde{\pi}_1, \tilde{\pi}_2$  be the permutations defined in Definition 1. For  $0 \leq z_1 \leq 2^{S_l} - 1$  and  $0 \leq z_2 \leq 2^{S_l} - 1$ ,*

$$\pi(z_1 + 2^{S_l} z_2) = \pi_1(z_1) + \pi_2(2^{S_l} z_2), \quad (2)$$

$$\pi(z_1 + 2^{S_l} z_2) = \tilde{\pi}_1(z_1) + 2^{S_l} \tilde{\pi}_2(z_2). \quad (3)$$

Moreover, if  $s_l = 1$ , then  $\pi = \pi_1$  and  $\pi_2$  is the identical permutation, which means

$$\pi(z_1 + 2^{m-1} z_2) = \pi(z_1) + 2^{m-1} z_2 \quad (4)$$

for  $0 \leq z_1 \leq 2^{m-1} - 1$  and  $z_2 = 0, 1$ . And (4) means  $\pi([0, n/2 - 1]) = [0, n/2 - 1]$  and  $\pi([n/2, n - 1]) = [n/2, n - 1]$ , that is, bits in the upper (lower) half branch remain in the upper (lower) half branch after permutation.

*Proof.* Since  $\pi = \pi_2 \circ \pi_1$ ,

$$\begin{aligned} & \pi(z_1 + 2^{S_l} z_2) \\ &= \pi_2 \circ \pi_1(z_1 + 2^{S_l} z_2) \\ &= \pi_2(\pi_1(z_1) + 2^{S_l} z_2) \\ &= \pi_1(z_1) + \pi_2(2^{S_l} z_2). \end{aligned}$$

So (2) is proved. (3) is from  $\pi_1(z_1) = \tilde{\pi}_1(z_1)$  and  $\pi_2(2^{S_l} z_2) = 2^{S_l} \tilde{\pi}_2(z_2)$ .  $\square$

**Remark 2.** Due to the block lower-triangular structure of  $M$ ,  $\pi = \pi_2 \circ \pi_1 = \pi_1 \circ \pi_2$ , where  $\pi_1$  only affects the first  $S_l$  bits and  $\pi_2$  only affects the last  $s_l$  bits. To be specific, permutation  $\pi$  can be decomposed into two steps.

Step 1 (the effect of  $\pi_1$ ): Divide  $[0, 2^m - 1]$  into  $2^{s_l}$  blocks  $[i2^{s_l}, (i+1)2^{s_l} - 1]$ ,  $0 \leq i \leq 2^{s_l} - 1$ , then apply the same permutation  $\tilde{\pi}_1$  to each block.

Step 2 (the effect of  $\pi_2$ ): Treat each block as a whole, apply  $\tilde{\pi}_2$  to  $2^{s_l}$  blocks.

This technique will help us decompose the polar code into shorter subcodes in Algorithm 1.

### C. Distinguishing SC-invariant automorphisms

Algorithm 1 determines whether affine automorphisms with the block lower-triangular structure  $\langle s_1, \dots, s_l \rangle$  commute with  $SC_{\mathcal{I}}$  iteratively. We claim that  $\pi = \varphi(M)$  commutes with  $SC_{\mathcal{I}}$  if and only if  $\text{DecAut}(s(M), \mathcal{I})$  outputs TRUE. Let  $\pi_1, \pi_2, \tilde{\pi}_1, \tilde{\pi}_2$  be the permutations defined in Definition 1. Note that  $s(\tilde{\pi}_1) = \langle s_1, \dots, s_{l-1} \rangle$ . We briefly describe procedures of Algorithm 1.

First, if  $l = 1$  (lines 5-7),  $\pi$  is SC-invariant if and only if the code belongs to Rate-0, single parity check (SPC), repetition (Rep) or Rate-1 codes [13].

If  $l \neq 1$ , we recursively determine whether  $\pi$  is SC-invariant. According to  $s_l$ , we consider two cases:

1)  $s_l = 1$  (lines 8-11), because  $\pi_2$  is the identical permutation,  $\pi$  is SC-invariant if and only if  $\tilde{\pi}_1$  commutes with  $C(\mathcal{I}(A_1))$  and  $C(\mathcal{I}(A_2))$ , i.e., the subcodes on upper half branch and lower half branch.

2)  $s_l > 1$  (lines 12-23), divide  $[0, 2^m - 1]$  into  $2^{s_l}$  blocks  $[i2^{s_l}, (i+1)2^{s_l} - 1]$ ,  $0 \leq i \leq 2^{s_l} - 1$ . In this case,  $\tilde{\pi}_2$  is not identical permutation, then  $\pi$  is SC-invariant only if all frozen bits belong to the first block or all information bits belong to the last block. Furthermore,  $\tilde{\pi}_1$  must commute with either the first subcode  $C(\mathcal{I}(A_1))$  (lines 13-14) or the last subcode  $C(\mathcal{I}(A_2))$  (lines 16-17) respectively.

---

### Algorithm 1 DecAut( $\langle s_1, \dots, s_l \rangle, \mathcal{I}$ )

---

**Input:** block lower-triangular structure  $\langle s_1, \dots, s_l \rangle$ , information set  $\mathcal{I}$

**Output:**  $a$  is a boolean value and  $a$  is TRUE if and only if automorphisms with the block lower-triangular structure  $\langle s_1, \dots, s_l \rangle$  commute with  $SC_{\mathcal{I}}$ .

```

1:  $m \leftarrow \sum_{i=1}^l s_i$ ;  $S_l \leftarrow \sum_{i=1}^{l-1} s_i$ ;
2:  $\mathcal{F} \leftarrow \{0, \dots, 2^m - 1\} / \mathcal{I}$ ;
3:  $A_1 \leftarrow \text{Ind}_m(a_{S_l+1} = 1, \dots, a_m = 1)$ ;
4:  $A_2 \leftarrow \text{Ind}_m(a_{S_l+1} = 0, \dots, a_m = 0)$ ;
5: if  $l = 1$  then
6:    $a \leftarrow (\mathcal{F} \subseteq A_1) \vee (\mathcal{I} \subseteq A_2)$ ;
7: else
8:   if  $s_l = 1$  then
9:      $a_1 \leftarrow \text{DecAut}(\langle s_1, \dots, s_{l-1} \rangle, \mathcal{I}(A_1))$ ;
10:     $a_2 \leftarrow \text{DecAut}(\langle s_1, \dots, s_{l-1} \rangle, \mathcal{I}(A_2))$ ;
11:     $a \leftarrow a_1 \wedge a_2$ ;
12:   else
13:     if  $\mathcal{F} \subseteq A_1$  then
14:        $a \leftarrow \text{DecAut}(\langle s_1, \dots, s_{l-1} \rangle, \mathcal{I}(A_1))$ ;
15:     else
16:       if  $\mathcal{I} \subseteq A_2$  then
17:          $a \leftarrow \text{DecAut}(\langle s_1, \dots, s_{l-1} \rangle, \mathcal{I}(A_2))$ ;
18:       else
19:          $a \leftarrow \text{FALSE}$ ;
20:       end if
21:     end if
22:   end if
23: end if

```

---

**Example 1.** Assume  $C(\mathcal{I})$  is a polar code with length  $n = 16$  and information set  $\mathcal{I} = \{3, 5, 6, 7, 9, 10, 11, 12, 13, 14, 15\}$ . It is clear that  $\text{Aut}(C(\mathcal{I})) = \text{BLTA}([4])$ . We can determine whether  $\varphi(M)$  with  $s(M) = \langle 3, 1 \rangle$  commutes with  $SC_{\mathcal{I}}$  by Algorithm 1. Since  $s_2 = 1$ , from lines 8-11,  $\text{DecAut}(\langle 3, 1 \rangle, \mathcal{I}) = \text{DecAut}(\langle 3 \rangle, \{3, 5, 6, 7\}) \wedge \text{DecAut}(\langle 3 \rangle, \{1, 2, 3, 4, 5, 6, 7\})$ . Next,  $\text{DecAut}(\langle 3 \rangle, \{3, 5, 6, 7\}) = \text{FALSE}$  and  $\text{DecAut}(\langle 3 \rangle, \{1, 2, 3, 4, 5, 6, 7\}) = \text{TRUE}$  from line 6. Therefore, we conclude that the algorithm will output FALSE so that  $\varphi(M)$  with  $s(M) = \langle 3, 1 \rangle$  does not commute with  $SC_{\mathcal{I}}$ .

In Theorem 1 and Theorem 2, we prove the sufficiency and necessity of Algorithm 1, respectively.

For convenience, denote by  $L_{i,t}$  and  $u_{i,t}$  the LLRs and hard decisions of the  $i$ -th node at stage  $t$  before permutation, and  $L'_{i,t}$  and  $u'_{i,t}$  the LLRs and hard decisions after permutation (see Section II-C).

**Theorem 1 (Sufficiency).** Let  $M$  be a block lower-triangular matrix with  $s(M) = \langle s_1, \dots, s_l \rangle$  and  $\pi = \varphi(M)$  is an automorphism of  $C(\mathcal{I})$  with length  $n = 2^m$ .  $\pi$  commutes with

$SC_{\mathcal{I}}$  if  $\text{DecAut}(s(M), \mathcal{I})$  outputs TRUE.

*Proof.* From Remark 1, assume  $M$  is an upper-triangular matrix. We prove the theorem by induction on  $l$ . If  $l = 1$ , the theorem holds since the condition implies the code is one of Rate-0, SPC, Rep or Rate-1 code, where SC decoding is equivalent to ML decoding, and thus invariant under permutations, and any permutation produces the same decoding result.

Let  $\pi_1, \pi_2, \tilde{\pi}_1, \tilde{\pi}_2$  be the permutations defined in Definition 1. For the induction step  $l - 1 \rightarrow l$ , There are two cases we need to consider. The first is  $s_l = 1$ . Divide  $[0, 2^{m-1}]$  into upper half branch  $[0, 2^{m-1} - 1]$  and lower half branch  $[2^{m-1}, 2^m - 1]$ . As mentioned in Remark 2,  $\tilde{\pi}_2$  is the identical permutation so bits in the upper or lower half branch remain in the same branch after permutation. Thus, the LLRs at stage  $m - 1$  of the upper and lower half branches are permuted by  $\tilde{\pi}_1$ , respectively. Then by induction,  $\tilde{\pi}_1$  is SC-invariant. It follows that  $\pi$  is SC-invariant.

Now let us discuss the proof in detail. First consider the upper half branch. Note that  $\pi(i) = \tilde{\pi}_1(i)$  for  $0 \leq i \leq 2^{m-1} - 1$ . (4) implies  $L'_{i+2^{m-1}, m} = L_{\pi(i+2^{m-1}), m} = L_{\tilde{\pi}_1(i)+2^{m-1}, m}$  and then

$$\begin{aligned} L_{\tilde{\pi}_1(i), m-1} &= f(L_{\tilde{\pi}_1(i), m}, L_{\tilde{\pi}_1(i)+2^{m-1}, m}) \\ &= f(L_{\pi(i), m}, L_{\pi(i)+2^{m-1}, m}) \\ &= f(L'_{i, m} + L'_{i+2^{m-1}, m}) = L'_{i, m-1}. \end{aligned}$$

Because that  $\text{DecAut}(\langle s_1, \dots, s_{l-1}, 1 \rangle, \mathcal{I})$  outputs TRUE implies that  $\text{DecAut}(\langle s_1, s_2, \dots, s_{l-1} \rangle, \mathcal{I}(\text{Ind}_m(a_m = 1)))$  outputs TRUE, by inductive hypothesis,

$$u'_{i, m-1} = u_{\tilde{\pi}_1(i), m-1} = u_{\pi(i), m-1}. \quad (5)$$

Next, we consider the lower half branch. For  $0 \leq i \leq 2^{m-1} - 1$

$$\begin{aligned} L_{\tilde{\pi}_1(i)+2^{m-1}, m-1} &= g(u_{\tilde{\pi}_1(i), m-1}, L_{\tilde{\pi}_1(i), m}, L_{\tilde{\pi}_1(i)+2^{m-1}, m}) \\ &= g(u'_{i, m-1}, L'_{i, m}, L'_{i+2^{m-1}, m}) \\ &= L'_{i+2^{m-1}, m-1}. \end{aligned}$$

Because that  $\text{DecAut}(\langle s_1, \dots, s_{l-1}, 1 \rangle, \mathcal{I})$  outputs TRUE implies that  $\text{DecAut}(\langle s_1, s_2, \dots, s_{l-1} \rangle, \mathcal{I}(\text{Ind}_m(a_m = 0)))$  outputs TRUE, by inductive hypothesis,

$$u'_{i+2^{m-1}, m-1} = u_{\tilde{\pi}_1(i)+2^{m-1}, m-1} = u_{\pi(i)+2^{m-1}, m-1}. \quad (6)$$

It follows from (5) and (6) that  $u_{\pi(i), m} = u'_{i, m}$ .

Now we turn to the case  $s_l > 1$ . In this case,  $\tilde{\pi}_2$  is not identical permutation, additional conditions are required to ensure SC-invariance of  $\pi$ . We divide  $[0, 2^m - 1]$  into  $2^{s_l}$  blocks  $\text{Ind}_m(a_{S_l+1} = c_{S_l+1}, \dots, a_m = c_m)$ . By lines 6-10 of Algorithm 1, either all frozen bits belong to the first block  $A_1 = \text{Ind}_m(a_{S_l+1} = 1, \dots, a_m = 1)$  or all information bits belong to the last block  $A_2 = \text{Ind}_m(a_{S_l+1} = 0, \dots, a_m = 0)$ .

1) If  $\mathcal{F} \subseteq A_1$ , from Lemma 2, we have  $L_{\pi_1(i), S_l} = L'_{i, S_l}$  for  $i \in A_1$ . Notice that  $\text{DecAut}(\langle s_1, s_2, \dots, s_l \rangle, \mathcal{I})$  outputs TRUE and  $\mathcal{F} \subseteq A_1$  imply that  $\text{DecAut}(\langle s_1, s_2, \dots, s_{l-1} \rangle, \mathcal{I}(A_1))$  outputs TRUE. Then by inductive hypothesis,  $u_{\pi_1(i), S_l} = u'_{i, S_l}$  for  $i \in A_1$ . Notice that  $\text{Ind}_m(a_{S_l+1} = c_{S_l+1}, \dots, a_m = c_m) \subseteq \mathcal{I}$

for  $a_{S_l+1}, \dots, a_m$  are not all one, then  $u_{i, S_l} = \text{sign}(L_{i, S_l})$  for  $i \notin A_1$ .

Then  $C(\mathcal{I})$  can be viewed as  $2^{S_l}$  independent length- $2^{S_l}$  SPC codes. Define  $A' = \text{Ind}_m(a_1 = c_1, \dots, a_{S_l} = c_{S_l})$  and  $\tilde{y} = (L'_{i, m})_{i \in A'} = (L_{\pi(i), m})_{i \in A'} = \tilde{\pi}_2(L_{\pi_1(i), m})_{i \in A'}$ , then  $(u'_{i, m})_{i \in A'} = \text{SC}_{\mathcal{I}'}(\tilde{y})$  where  $\mathcal{I}' = \{1, \dots, 2^{S_l} - 1\}$  and the first bit is frozen to  $u'_{z_1, S_l}$  with  $z_1 = (a_1, \dots, a_{S_l}, 1, \dots, 1)^T$ . That is,  $(u'_{i, m})_{i \in A'}$  can be decoded as a length- $2^{S_l}$  SPC code with LLR vector  $\tilde{y}$ .

Since  $\tilde{\pi}_2$  commutes with  $\mathcal{I}'$ , we have

$$\begin{aligned} (u'_{i, m})_{i \in A'} &= \text{SC}_{\mathcal{I}'}(\tilde{y}) = \text{SC}_{\mathcal{I}'}(\tilde{\pi}_2(L_{\pi_1(i), m})_{i \in A'}) \\ &= \tilde{\pi}_2(\text{SC}_{\mathcal{I}'}(L_{\pi_1(i), m})_{i \in A'}) = \tilde{\pi}_2(u_{\pi_1(i), m})_{i \in A'} \\ &= (u_{\pi(i), m})_{i \in A'}, \end{aligned}$$

thus  $u'_{i, m} = u_{\pi(i), m}$ .

2) If  $\mathcal{I} \subseteq A_2$  we have  $u_{i, S_l} = u'_{i, S_l} = 0$  for  $i \notin A_2$ . Thus,

$$u_{i, m} = u_{j, S_l}; u'_{i, m} = u'_{j, S_l}. \quad (7)$$

for  $j \in A_2$  and  $j \equiv i \pmod{2^{S_l}}$ . From Lemma 2,  $L_{\pi_1(j), S_l} = L'_{j, S_l}$  for  $j \in A_2$ . Notice that  $\text{DecAut}(\langle s_1, s_2, \dots, s_l \rangle, \mathcal{I})$  outputs TRUE and  $\mathcal{I} \subseteq A_2$  imply that  $\text{DecAut}(\langle s_1, s_2, \dots, s_{l-1} \rangle, \mathcal{I}(A_2))$  outputs TRUE. By inductive hypothesis,

$$u_{\pi_1(j), S_l} = u'_{j, S_l}. \quad (8)$$

Then

$$u'_{i, m} = u'_{j, S_l} = u_{\pi_1(j), S_l} = u_{\pi(i), m},$$

where  $j \equiv i \pmod{2^{S_l}}$  and  $j \in A_2$ . Here the first equation is from (7), the second equation is from (8), and the last is because of (7) and  $\pi_1(j) \equiv \pi(j) \equiv \pi(i) \pmod{2^{S_l}}$  from Lemma 2.  $\square$

The next lemma allows us to claim an automorphism is not SC-invariant by decomposing the automorphism on the upper and lower half branches even if  $s_l \neq 1$ . It will help us prove the necessity.

**Lemma 3.** *Let  $C(\mathcal{I})$  be a decreasing monomial code with length  $n = 2^m$ .  $\pi = \varphi(M)$  is an automorphism of  $C(\mathcal{I})$ , where  $M$  is an upper-triangular matrix. Let  $A_i = \text{Ind}_m(a_m = i)$  and  $\mathcal{I}_i = \mathcal{I}(A_i)$ ,  $i = 0, 1$ , denote  $\pi' = \varphi(M([1, m-1], [1, m-1]))$ , then  $\pi$  commutes with  $SC_{\mathcal{I}}$  implies  $\pi'$  commutes with  $SC_{\mathcal{I}_1}$  and  $SC_{\mathcal{I}_0}$ , i.e.,  $\pi'$  commutes with the subcodes on upper and lower half branches.*

*Proof.* If  $\pi'$  does not commute with  $SC_{\mathcal{I}_1}$ , because  $\pi'' \triangleq \pi|_{A_1} = (M([1, m-1], [1, m-1]), M([1, m-1], m))$ ,  $\pi''$  does not commute with  $SC_{\mathcal{I}_1}$  as well. So there exists  $y \in \mathbb{R}^{2^{m-1}}$  such that  $\pi''(\text{SC}_{\mathcal{I}_1}(y)) \neq \text{SC}_{\mathcal{I}_1}(\pi''(y))$ .

Now we can construct an example from  $y$  to show  $\pi$  does not commute with  $SC_{\mathcal{I}}$ . To be specific, let  $(L_{i, m})_{i \in A_1} = y$  and  $(L_{i, m})_{i \in A_0} = +\infty$ , then  $L_{i, m-1} = f(L_{i, m}, +\infty) = L_{i, m}$  for  $i \in A_1$ . Therefore,  $(L_{i, m-1})_{i \in A_1} = y$  and  $(L'_{i, m-1})_{i \in A_1} = \pi''(y)$ . Since  $\pi''(\text{SC}_{\mathcal{I}_1}(y)) \neq \text{SC}_{\mathcal{I}_1}(\pi''(y))$ , we have for some  $j$

$$u_{\pi''(j), m-1} \neq u'_{j, m-1}. \quad (9)$$

For  $i \in A_1$ ,  $L_{i+2^{m-1}, m-1} = g(u_{i, m-1}, L_{i, m}, +\infty) = +\infty$ . Similarly,  $L'_{i+2^{m-1}, m-1} = +\infty$ . Thus

$$u_{i+2^{m-1}, m-1} = u'_{i+2^{m-1}, m-1} = 0. \quad (10)$$

Together with (9) and (10),  $u_{\pi(j), m} \neq u'_{j, m}$  for some  $j$ .

$\pi'$  commutes with  $\text{SC}_{\mathcal{I}_0}$  can be proved similarly when  $(L_{i, m})_{i \in A_1} = \varepsilon$  and  $(L_{i, m})_{i \in A_0} = y$ , where  $\varepsilon$  is positive and small enough.  $\square$

The next lemma proves two special cases of the necessity by decomposing the permutation on the subcodes consisting of odd and even indices.

**Lemma 4.** *Let  $M$  be a block lower-triangular matrix with  $s(M) = \langle 1, 1, \dots, 1, s_l \rangle$  where  $s_l = 2$  or  $3$ .  $\pi = \varphi(M)$  is an automorphism of  $C(\mathcal{I})$  with length  $n = 2^m$ . Then  $\pi$  commutes with  $C(\mathcal{I})$  only if  $\mathcal{F} \subseteq \text{Ind}_m(a_{S_l+1} = 1, \dots, a_m = 1)$  or  $\mathcal{I} \subseteq \text{Ind}_m(a_{S_l+1} = 0, \dots, a_m = 0)$ .*

*Proof.* We inducted on  $m$ , if  $m = 2, 3$ , the lemma can be proved by exhaustive search. For the induction step  $m-1 \rightarrow m$ , assume  $s(M) = \langle 1, 1, \dots, 1, 2 \rangle$ . Let  $\mathcal{I}$  be an information set such that  $\mathcal{F} \not\subseteq A_1 = \text{Ind}_m(a_{m-1} = 1, a_m = 1)$  and  $\mathcal{I} \not\subseteq A_2 = \text{Ind}_m(a_{m-1} = 0, a_m = 0)$ . Divide  $\mathcal{I}$  into two information sets  $\mathcal{I}_1 = \mathcal{I}(\text{Ind}_m(a_1 = 1))$  on the even bits and  $\mathcal{I}_2 = \mathcal{I}(\text{Ind}_m(a_1 = 0))$  on the odd bits, then  $\mathcal{F}_1 = \text{Ind}_m(a_1 = 1) - \mathcal{I}_1$  and  $\mathcal{F}_2 = \text{Ind}_m(a_1 = 0) - \mathcal{I}_2$ .

We are going to show that at least one of  $\mathcal{I}_1$  and  $\mathcal{I}_2$  does not satisfy the condition. First,  $\mathcal{F}_1 \not\subseteq A_1$ , since  $\mathcal{F}_1 \subseteq A_1$  implies  $\mathcal{F}_2 \subseteq A_1$  by decreasing property, which is contradictory against  $\mathcal{F} \not\subseteq A_1$ . Similarly,  $\mathcal{I}_2 \not\subseteq A_2$ .

We claim that one of  $\mathcal{F}_2 \not\subseteq A_1$  and  $\mathcal{I}_1 \not\subseteq A_2$  must hold, since otherwise  $\text{Ind}_m(a_1 = 0, a_{m-1} = 1, a_m = 0) \subseteq \mathcal{I}$  and  $\text{Ind}_m(a_1 = 1, a_{m-1} = 1, a_m = 0) \subseteq \mathcal{F}$ . Since  $m > 4$ ,  $\text{Ind}_m(a_1 = 0, a_2 = 1, a_{m-1} = 1, a_m = 0) \subseteq \mathcal{I}$  and  $\text{Ind}_m(a_1 = 1, a_2 = 0, a_{m-1} = 1, a_m = 0) \subseteq \mathcal{F}$ , which are contradictory against  $\mathcal{I}$  is a decreasing set when  $m \geq 4$ .

Now we are going to construct a counter-example by induction. Assume  $\mathcal{I}_1 \not\subseteq A_2$ , denote  $\pi' = \varphi(M([2, m], [2, m]))$ . From inductive hypothesis, there exists some  $\tilde{y} \in \mathbb{R}^{2^{m-1}}$  such that  $\pi'(\text{SC}_{\mathcal{I}_1}(\tilde{y})) \neq \text{SC}_{\mathcal{I}_1}(\pi'(\tilde{y}))$ , which implies  $\varphi(M)$  does not commute with  $\text{SC}_{\mathcal{I}}$  by setting  $(L_{i, m})_{i \in \text{Ind}_m(a_1=1)} = \tilde{y}$  and  $L_{i, m} = +\infty$  otherwise. If  $\mathcal{F}_2 \not\subseteq A_1$ , denote  $(L_{i, m})_{i \in \text{Ind}_m(a_1=0)} = \tilde{y}$  and  $L_{i, m} = \varepsilon$  where  $\varepsilon$  is positive and small enough otherwise.

If  $s(M) = \langle 1, 1, \dots, 1, 3 \rangle$ , the proof is similar if we take  $A_1 = \text{Ind}_m(a_{m-2} = 1, a_{m-1} = 1, a_m = 1)$  and  $A_2 = \text{Ind}_m(a_{m-2} = 0, a_{m-1} = 0, a_m = 0)$ .  $\square$

Now we are ready to prove Theorem 2.

**Theorem 2 (Necessity).** *Let  $M$  be a block lower-triangular matrix with  $s(M) = \langle s_1, \dots, s_l \rangle$  and  $\pi = \varphi(M)$  is an automorphism of  $C(\mathcal{I})$  with length  $n = 2^m$ .  $\pi$  commutes with  $\text{SC}_{\mathcal{I}}$  only if  $\text{DecAut}(s(M), \mathcal{I})$  outputs TRUE.*

*Proof.* From Remark 1, assume  $M$  is an upper-triangular matrix. We prove the theorem by induction on  $m$ . if  $m \leq 3$ ,

it can be proved by computer search. Assume the theorem holds for  $m' \leq m-1$ . Define  $A_1 = \text{Ind}_m(a_m = 1)$  and  $A_0 = \text{Ind}_m(a_m = 0)$ . Cases are classified according to  $s_l$ .

If  $s_l = 1$ , the theorem can be proved by Lemma 3.

If  $s_l = 2$  or  $3$ , Let  $\pi_1, \pi_2, \tilde{\pi}_1, \tilde{\pi}_2$  be the permutations defined in Definition 1. Divide  $[0, 2^m - 1]$  into  $2^{s_l}$  blocks  $\text{Ind}_m(a_{S_l+1} = c_{S_l+1}, \dots, a_m = c_m)$ . Denote  $\mathcal{I}' = \mathcal{I}(\text{Ind}_m(a_{S_l+1} = c_{S_l+1}, \dots, a_m = c_m))$ . Since  $\pi$  is SC-invariant, repeatedly applying Lemma 3 reveals that  $\tilde{\pi}_1$  commutes with  $\text{SC}_{\mathcal{I}'}$ . By Theorem 1,  $\pi_1$  commutes with  $\text{SC}_{\mathcal{I}'}$ . Therefore,  $\pi_2 = \pi_1^{-1} \circ \pi$  commutes with  $\text{SC}_{\mathcal{I}'}$ . Then the theorem can be proved by Lemma 4.

If  $s_l \geq 4$ , without loss of generality, assume  $s(M([1, m-1], [1, m-1])) = \langle s_1, \dots, s_{l-1}, s_l - 1 \rangle$  and  $M(m, [1, m-1]) = 0$ . Define  $\pi' = \varphi(M([1, m-1], [1, m-1]))$ . (This can be achieved by transformations in Lemma 1.) From Lemma 3,  $\pi$  commutes with  $\text{SC}_{\mathcal{I}}$  only if  $\pi'$  commutes with  $\text{SC}_{\mathcal{I}(A_i)}$  for  $i = 0, 1$ .

From inductive hypothesis, for all  $i = 0, 1$ , one of  $\mathcal{F}(A_i) \subseteq \text{Ind}_m(a_{S_l+1} = 1, \dots, a_{m-1} = 1, a_m = i)$  and  $\mathcal{I}(A_i) \subseteq \text{Ind}_m(a_{S_l+1} = 0, \dots, a_{m-1} = 0, a_m = i)$  holds. Now we are going to prove one of  $\mathcal{F} \subseteq \text{Ind}_m(a_{S_l+1} = 1, \dots, a_{m-1} = 1, a_m = 1)$  and  $\mathcal{I} \subseteq \text{Ind}_m(a_{S_l+1} = 0, \dots, a_{m-1} = 0, a_m = 0)$  must hold. We consider the following three cases:

1) If  $\mathcal{I}(A_1) = \emptyset$ , then  $A_1 \subseteq \mathcal{F}$ . Since  $\varphi(M)$  with  $s(M) = \langle s_1, \dots, s_l \rangle$  is an automorphism of  $C(\mathcal{I})$ , for any permutations  $\rho \in \text{sym}(m)$  that permutes  $[S_j + 1, S_{j+1}]$  to  $[S_j + 1, S_{j+1}]$  for  $1 \leq j \leq l$ ,  $(a_1, \dots, a_m) \in \mathcal{I}$  is equal to  $(a_{\rho(1)}, \dots, a_{\rho(m)})^T \in \mathcal{I}$ . Therefore,  $A_1 \subseteq \mathcal{F}$  implies  $[n/2, n - 2^{S_l} - 1] \subseteq \mathcal{F}$ . Thus,  $\mathcal{I} \subseteq \text{Ind}_m(a_{S_l+1} = 0, \dots, a_{m-1} = 0, a_m = 0)$  must hold.

2) If  $\mathcal{F}(A_0) = \emptyset$ , similarly,  $\mathcal{F} \subseteq \text{Ind}_m(a_{S_l+1} = 1, \dots, a_{m-1} = 1, a_m = 1)$  must hold.

3) If  $\mathcal{I}(A_1) \neq \emptyset$  and  $\mathcal{F}(A_0) \neq \emptyset$ . By properties of affine automorphism group,  $\mathcal{I}(A_1) \neq \emptyset$  implies  $\mathcal{I}(A_0) \not\subseteq \text{Ind}_m(a_{S_l+1} = 0, \dots, a_{m-1} = 0, a_m = 0)$ . Thus

$$\mathcal{F}(A_0) \subseteq \text{Ind}_m(a_{S_l+1} = 1, \dots, a_{m-1} = 1, a_m = 0).$$

Similarly,

$$\mathcal{I}(A_1) \subseteq \text{Ind}_m(a_{S_l+1} = 0, \dots, a_{m-1} = 0, a_m = 1).$$

Then  $\text{Ind}_m(a_{m-3} = 0, a_{m-2} = 1, a_{m-1} = 1, a_m = 0) \subseteq \mathcal{I}$  and  $\text{Ind}_m(a_{m-3} = 1, a_{m-2} = 0, a_{m-1} = 0, a_m = 1) \subseteq \mathcal{F}$ , which is contradictory against automorphism group.  $\square$

From Algorithm 1, SC-invariance of  $\varphi(M)$  only depends on the block lower-triangular structure of  $M$ . Thus, we can prove the following theorem.

**Theorem 3.**  $[\mathbb{1}]_{\mathcal{I}}$  is in the form of BLTA.

*Proof.* Let  $M$  be a block lower-triangular matrix with  $s(M) = \langle s_1, \dots, s_l \rangle$  satisfying  $\pi = \varphi(M)$  commutes with  $\text{SC}_{\mathcal{I}}$  and  $l$  is as small as possible. Then  $\text{BLTA}([s_1, \dots, s_l]) \subseteq [\mathbb{1}]_{\mathcal{I}}$ . Now assume  $\text{BLTA}([s'_1, \dots, s'_k]) \subset [\mathbb{1}]_{\mathcal{I}}$  but  $\text{BLTA}([s'_1, \dots, s'_k]) \not\subseteq \text{BLTA}([s_1, \dots, s_l])$ . Then there must exist some  $i, j$  such that  $S'_i < S_j < S'_{i+1}$ . Let  $M_1$  be a permutation matrix which

$(n, K)$	$\mathcal{I}_{\min}$	affine automorphism group	$[\mathbf{1}]_{\mathcal{I}}$	SC-invariant permutations in [10]	SC-invariant permutations in this paper
(256, 128)	{31, 57}	BLTA([3, 5])	BLTA([3, 1, 1, 1, 1, 1])	$3 \times 2^{28}$	$21 \times 2^{28}$
(128, 85)	{23, 25}	BLTA([3, 1, 3])	BLTA([3, 1, 1, 1, 1])	$3 \times 2^{21}$	$21 \times 2^{21}$
(64, 32)	{24}	BLTA([3, 3])	BLTA([3, 2, 1])	$3 \times 2^{15}$	$63 \times 2^{15}$

Table 1: The number of SC-invariant permutations for certain codes

---

**Algorithm 2** DecGroup( $\mathcal{I}, m$ )

---

**Input:** the information sets  $\mathcal{I}$ , the code dimension  $m$ .

**Output:**  $s = [s_1, \dots, s_l]$ ; # BLTA( $[s_1, \dots, s_l]$ ) =  $[\mathbf{1}]_{\mathcal{I}}$

```

1:  $\mathcal{F} \leftarrow \{0, \dots, 2^m - 1\} / \mathcal{I}$ ;
2: if  $m = 0$  then
3:    $s = []$ ;
4:   return;
5: end if
6: for  $t = m; t \geq 2; t --$  do
7:    $A_1 \leftarrow \text{Ind}_m(a_{m-t+1} = 1, \dots, a_m = 1)$ ;
8:    $A_2 \leftarrow \text{Ind}_m(a_{m-t+1} = 0, \dots, a_m = 0)$ ;
9:   if  $\mathcal{F} \subseteq A_1$  then
10:     $s \leftarrow [\text{DecGroup}(\mathcal{I}(A_1), m - t), t]$ ;
11:    return;
12:   else
13:     if  $\mathcal{I} \subseteq A_2$  then
14:        $s \leftarrow [\text{DecGroup}(\mathcal{I}(A_2), m - t), t]$ ;
15:       return;
16:     end if
17:   end if
18: end for
19:  $s' \leftarrow [\text{DecGroup}(\mathcal{I}(\text{Ind}_m(a_m = 1), m - 1), 1)]$ ;
20:  $s'' \leftarrow [\text{DecGroup}(\mathcal{I}(\text{Ind}_m(a_m = 0), m - 1), 1)]$ ;
21:  $s \leftarrow \text{Gro}(s', s'')$ ; # BLTA( $s$ ) = BLTA( $s'$ )  $\cap$  BLTA( $s''$ ).

```

---

permutes  $S'_i$  and  $S_j$  and keeps the other positions invariable, then  $\varphi(M_1) \in \text{BLTA}([s'_1, \dots, s'_k])$ . However,  $s(M_1M) = \langle s_1, \dots, s_{j-2}, s_{j-1} + s_j, s_{j+1}, \dots, s_l \rangle$  and  $\varphi(M_1M) \in [\mathbf{1}]_{\mathcal{I}}$ , which is contradictory against that  $l$  is as small as possible.  $\square$

In Algorithm 1, we determine whether an affine automorphism commutes with  $\text{SC}_{\mathcal{I}}$ . With Algorithm 2, we further determine the complete SC-invariant affine automorphism group  $[\mathbf{1}]_{\mathcal{I}} = \text{BLTA}(\text{DecGroup}(\mathcal{I}, m))$ .

Without loss of generalization, assume  $[\mathbf{1}]_{\mathcal{I}} = \text{BLTA}([s_1, \dots, s_l])$ . We first determine  $s_l$ , then  $[s_1, \dots, s_{l-1}]$  can be obtained by calling the algorithm recursively.

First,  $s_l$  is determined by the loop in line 6. For  $2 \leq t \leq m$ , divide  $[0, 2^m - 1]$  to  $2^t$  blocks. We have  $s_l = t$  if and only if  $t$  is the largest integer such that all frozen bits belong to the first block  $A_1 = [0, 2^{m-t} - 1]$  (line 9) or all information bits belong to the last block  $A_2 = [2^m - 2^{m-t}, 2^m - 1]$  (line 13). If for all  $2 \leq t \leq m$ , the above conditions are not satisfied, we have  $s_l = 1$ .

If  $s_l \geq 2$ ,  $[s_1, \dots, s_{l-1}]$  can be recursively obtained by calling the algorithm with  $\mathcal{I}(A_1)$  (line 10) when  $\mathcal{F} \subseteq A_1$  or

$\mathcal{I}(A_2)$  (line 14) when  $\mathcal{I} \subseteq A_2$ . If  $s_l = 1$ ,  $\text{BLTA}[s_1, \dots, s_{l-1}]$  is the intersection of the SC-invariant affine automorphism groups of subcodes on the upper and lower half branches (lines 19-21). In line 21,  $\text{Gro}(s', s'')$  output the array  $s$  satisfying  $\text{BLTA}(s) = \text{BLTA}(s') \cap \text{BLTA}(s'')$ . Such  $s$  exists and can be found by the following lemma.

**Lemma 5.** *The intersection of two BLTA groups is in the form of BLTA.*

*Proof.* We are going to find the BLTA group which is the intersection of  $\text{BLTA}(s')$  and  $\text{BLTA}(s'')$ . Let  $\{S_t\} = \{S'_t\} \cup \{S''_t\}$ , and  $s$  is induced by  $\{S_t\}$ , that is,  $s_t = S_{t+1} - S_t$ . Next we are going to prove  $\text{BLTA}(s) = \text{BLTA}(s') \cap \text{BLTA}(s'')$ .

It is clear that  $\text{BLTA}(s) \subseteq \text{BLTA}(s')$  and  $\text{BLTA}(s) \subseteq \text{BLTA}(s'')$ . Therefore, we only need to prove  $\text{BLTA}(s') \cap \text{BLTA}(s'') \subseteq \text{BLTA}(s)$ . Assume  $(M, b) \in \text{BLTA}(s') \cap \text{BLTA}(s'')$ . Now we consider  $M(j, k)$  for  $S_i + 1 \leq j \leq S_{i+1}$  and  $S_{i+1} + 1 \leq k \leq m$ . Without loss of generality, assume  $S_i = S'_i$ . By the construction of  $\{S_t\}$ , we have  $S_{i+1} \leq S'_{i+1}$ . Then  $M(j, k) = 0$  since  $(M, b) \in \text{BLTA}(s')$ . Therefore,  $(M, b) \in \text{BLTA}(s)$ .  $\square$

**Remark 3.** Algorithm 2 selects each  $s_i$  as its largest possible value such that Algorithm 1 will not output FALSE, so it will output the complete SC-invariant affine automorphism group. Otherwise, if there exists another SC-invariant automorphism not in the output group, from Theorem 3, there will be a larger SC-invariant BLTA automorphism group with some larger  $s_i$ , which is a contradiction. Since the time complexity of one iteration is  $O(m)$ , the complexity of Algorithm 2 is  $O(m2^m) = O(n \log n)$ .

**Example 2.** We now determine the complete SC-invariant affine automorphism group of  $C(\mathcal{I})$  in Example 1 by Algorithm 2. For all  $2 \leq t \leq 4$ , the conditions in line 9 and line 13 are not satisfied, so the last number of  $s$  is 1. Then we call  $\text{DecGroup}(\{3, 5, 6, 7\}, 3)$  and  $\text{DecGroup}(\{1, 2, 3, 4, 5, 6, 7\}, 3)$ .  $\text{DecGroup}(\{3, 5, 6, 7\}, 3)$  will output  $[2, 1]$  and  $\text{DecGroup}(\{1, 2, 3, 4, 5, 6, 7\}, 3)$  will output  $[3]$ . Then  $s = \text{Gro}([2, 1, 1], [3, 1]) = [2, 1, 1]$ . Therefore, the complete SC-invariant affine automorphism group of  $C(\mathcal{I})$  is  $\text{BLTA}([2, 1, 1])$ .

#### IV. SIMULATION

Fig. 2 shows the block error rate (BLER) performance of the (256,128) polar code studied in [7] and [10]. The code is generated by  $\mathcal{I}_{\min} = \{31, 57\}$  and has affine automorphism group  $\text{BLTA}([3, 5])$ . In this case,  $[\mathbf{1}]_{\mathcal{I}} = \text{BLTA}([3, 1, 1, 1, 1, 1])$ , and it is shown that all the automorphisms in  $\text{BLTA}([3, 1, 1, 1, 1, 1])$  are futile in AE-SC decoding.

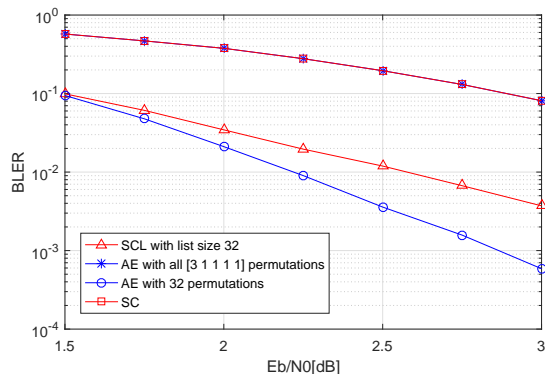


Fig. 2: Performance of a (256,128) polar code

Since the complete SC-invariant affine automorphisms are determined, the number of equivalent classes can be reduced from 68355 [10] to 9765.

Table 1 compares the number of SC-invariant affine automorphisms found in this paper with BLTA([2, 1..., 1]). Under several code constructions, the SC-invariant automorphism group can be larger than BLTA([2, 1..., 1]), which benefits applications requiring SC-invariant automorphisms.

## V. CONCLUSION

In this paper, we determine and prove the complete SC-invariant affine automorphisms for any specific decreasing polar code, which form a BLTA group. Compared to previous works, more SC-invariant affine automorphisms can be found according to our results. It helps us remove redundant permutations in AE-SC decoding and contributes to other applications requiring SC-invariant automorphisms.

## REFERENCES

- [1] E. Arıkan, "Channel Polarization: A Method for Constructing Capacity-Achieving Codes for Symmetric Binary-Input Memoryless Channels," in *IEEE Transactions on Information Theory*, vol. 55, no. 7, pp. 3051-3073, July 2009.
- [2] I. Tal and A. Vardy, "List Decoding of Polar Codes," in *IEEE Transactions on Information Theory*, vol. 61, no. 5, pp. 2213-2226, May 2015.
- [3] K. Niu and K. Chen, "CRC-Aided Decoding of Polar Codes," in *IEEE Communications Letters*, vol. 16, no. 10, pp. 1668-1671, October 2012.
- [4] N. Doan, S. A. Hashemi, M. Mondelli and W. J. Gross, "On the Decoding of Polar Codes on Permuted Factor Graphs," 2018 *IEEE Global Communications Conference (GLOBECOM)*, 2018, pp. 1-6.
- [5] M. Geiselhart, A. Elkelesh, M. Ebada, S. Cammerer and S. t. Brink, "Automorphism Ensemble Decoding of Reed-Muller Codes," in *IEEE Transactions on Communications*, vol. 69, no. 10, pp. 6424-6438, Oct. 2021.
- [6] M. Bardet, V. Dragoi, A. Otmani and J. -P. Tillich, "Algebraic properties of polar codes from a new polynomial formalism," 2016 *IEEE International Symposium on Information Theory (ISIT)*, 2016, pp. 230-234.
- [7] M. Geiselhart, A. Elkelesh, M. Ebada, S. Cammerer and S. ten Brink, "On the Automorphism Group of Polar Codes," 2021 *IEEE International Symposium on Information Theory (ISIT)*, 2021, pp. 1230-1235.
- [8] Y. Li, H. Zhang, R. Li, J. Wang, W. Tong, G. Yan, Z. Ma, (2021). The Complete Affine Automorphism Group of Polar Codes. arXiv preprint arXiv:2103.14215.
- [9] C. Pillet, V. Bioglio and I. Land, "Polar Codes for Automorphism Ensemble Decoding," 2021 *IEEE Information Theory Workshop (ITW)*, 2021, pp. 1-6.

- [10] C. Pillet, V. Bioglio and I. Land, (2021). Classification of Automorphisms for the Decoding of Polar Codes. arXiv preprint arXiv:2110.14438.
- [11] H. Luo et al., "Analysis and Application of Permuted Polar Codes," *IEEE Global Communications Conference (GLOBECOM)*, Abu Dhabi, United Arab Emirates, 2018, pp. 1-5.
- [12] C. Schürch, "A partial order for the synthesized channels of a polar code," 2016 *IEEE International Symposium on Information Theory (ISIT)*, 2016, pp. 220-224.
- [13] G. Sarkis, P. Giard, A. Vardy, C. Thibault and W. J. Gross, "Fast Polar Decoders: Algorithm and Implementation," in *IEEE Journal on Selected Areas in Communications*, vol. 32, no. 5, pp. 946-957, May 2014.